Mesh evaluation

The estimation of the spacing for the mesh along the y direction is performed through one of the utilities that can be found inside the folder `utilities/Stretching Mesh`. For a channel flow case, the default function of Incompact3d can be used, `stretching_parameter_channel.f90`.

For a temporal TBL case instead, a new function was developed, starting from the stretching subroutine that can be found inside the original solver of Incompact3d. This new function is a Python script, `mesh_evaluation.py`. This function allows to estimate the mesh size by introducing the following inputs:

- Number of points: n_x,n_y,n_z.
- Domain dimensions: L_x,L_y,L_z.
- Stretching parameter: β.
- Skin friction coefficient: c_f.
- Kinematic viscosity: ν.
- Velocity of the wall: U_w.
- Time step: Δt.
- Tripping wire diameter: D.

And it produces the following outputs:

- Non-dimensional domain dimensions: L_x^+,L_y^+,L_z^+.
- CFL, Péclet and Numerical Fourier: $Pé$, Co, $\mathcal{D}$.
- Mesh size at the first element near the wall: Δy_1^+.
- Mesh size at the last element away from the wall: Δy_n^+.
- Mesh spacings in x and z directions: Δx^+, Δz^+.
- Aspect ratios (ARs) of grid elements at bottom and top walls in x and z directions: AR_{x_1}, AR_{x_n}, AR_{z_1}, AR_{z_n}.
- Number of mesh nodes in the viscous sublayer and in the initial shear layer.
- Approximate and calculated values of initial shear layer thickness θ_{sl}.

It is worth noticing that in a temporal BL simulation, the peak c_f value constraints the height of the first cell at the wall Δy_1, as in standard CFD simulations. However, the decrease of c_f along the simulation (and thus the increase in viscous length δ_ν) imposes a constraint for the domain dimensions L_x, L_y, L_z since they appear progressively "smaller" (their non-dimensional counterparts decreases). Too low values of L_x^+,L_y^+,L_z^+ must be avoided in order to do not enforce a too strong periodicity in the turbulent structures at the wall.